Two applications of omnidirectional vision: visual-servo of serial and parallel robots

Youcef Mezouar[1] and Hicham Hadj Abdelkader[2] and Omar tahri[1]

[1] LASMEA, Aubiere F-63177, France
[2] INRIA, 06902 Sophia Antipolis, France

Abstract. In this paper we consider the problem of controlling a serial or a parallel robot from the projection of 3D straight lines in the image plane of central catadioptric systems. A generic central catadioptric interaction matrix for the projection of 3D straight lines is derived using an unifying imaging model valid for an entire class of cameras. This result is exploited to design an image-based control law which allows us to control the six degree of freedom of a robotic arm. Then the projected lines are exploited to design a control scheme for a parallel robot by observing the platform's legs. These two applications of omnidirectional vision are substantiated by experimental results with a six d.o.f eye-to-hand system and a Gough-Steward platform.

1 Introduction

Many applications in vision-based robotics, such as mobile robot localisation [1] and navigation [2], can benefit from a panoramic field of view provided by omnidirectional cameras. In the literature, there have been several methods proposed for increasing the field of view of cameras systems [3]. One effective way is to combine mirrors with conventional imaging system. The obtained sensors are referred to as catadioptric imaging systems. The resulting imaging systems have been termed central catadioptric when a single projection center describes the world-image mapping. From a theoretical and practical point of view, a single center of projection is a desirable property for an imaging system [4]. Baker and Nayar [4] derive the entire class of catadioptric systems with a single viewpoint.

Vision-based control schemes are flexible and effective methods to control robot motions from visual data [5]. Clearly, these applications can benefit from such sensors since they naturally overcome the visibility constraint. As an example, let us consider the problem of controlling the Gough-Stewart platform shown in Figure 1 using a vision sensor. At this aim, [6, 7] propose respectively image-based and position-based visual-servo schemes by directly observing the platform legs with a classical perspective camera. Unfortunately, to position adequately the camera to observe simultaneously all the platform legs is a complex task. In [6, 7], the camera was positioned in front of the platform (see Fig 1.(a)). In this case, the legs in the front of the platform are closer to the camera than the ones in the back. As a consequence, the extraction of the image features

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 182-193

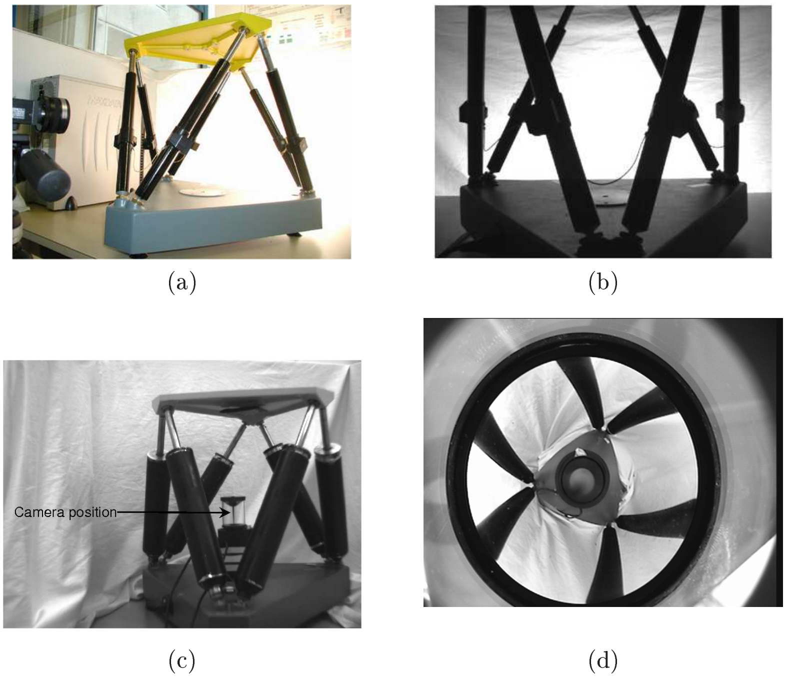

(a) (b) (c) (d)

Fig. 1. A Gough-Stewart platform observed by a classical perspective camera: (a) camera position with respect to the platform, (b) image of the legs. A Gough-Stewart platform observed by an omnidirectional camera: (c) camera position with respect to the platform, (d) image of the legs

lying on legs in the back will be less robust. Furthermore, large parts of the legs in the back are occluded by the front legs (see Fig 1.(b)) and full occlusions can happen. This is an important drawback since the vision based control assumes that all legs can be observed during the servoing task. A first solution to address this issue could be to employ a system made of multiple cameras. However, in this case, data provided by each camera must be synchronized and the multi-camera system calibrated. A second and simpler solution, whose first results were presented in [8], consists in positioning a single omnidirectional camera at the platform center (see Figure 1.(c)). In such a way, all the legs can be simultaneously observed in a panoramic view and potential occlusions can not occur (see Figure 1.(d)). Clearly, visual servoing of the Gough-Stewart platform will thus benefit from the enhanced field of view provided by an omnidirectional camera. More generally, visual servoing schemes make assumptions on the link between the initial, current and desired images since they require correspondences between the features extracted from the initial image with those obtained from the desired one. These measures are then tracked during the camera (and/or the object) motion. If one of these steps fails, then the task can not be achieved.

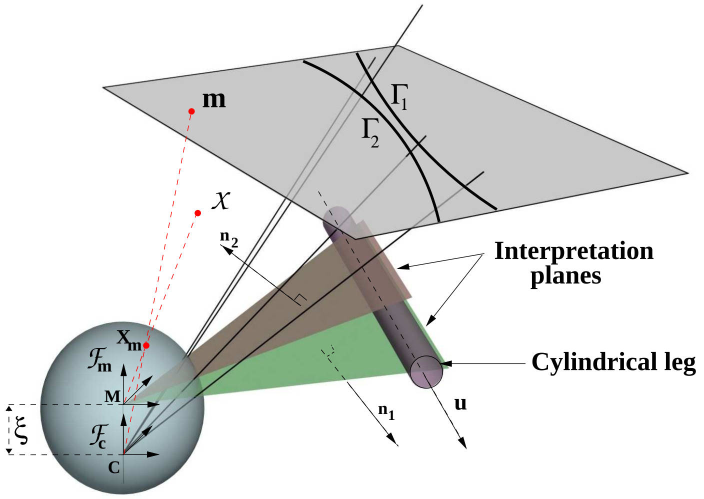


Fig. 2. Projection of a cylindrical leg onto the image plane

Typical cases of failure arise when matching joint image features is impossible (for example when no joint feature belongs to initial and desired images) or when some parts of the visual features get out of the field of view during the servoing. The use of omnidirectional vision should thus significantly reduce the case of failure. However, omnidirectional images exhibit supplementary difficulties compared to conventional perspective image (for example the projection of a line is no more a line but a conic curve). This paper is mainly concerned with the use of projected lines extracted from central catadioptric images as input to a visual servoing control loop. Two applications are described. The first one concerns the control of serial robots while the second one concerns the control of a parallel robot (namely a Gough-Stewart platform).

2 Modeling

2.1 Camera model

Central imaging systems can be modeled using two consecutive projections: spherical projection then perspective one. This geometric formulation called unified model has been proposed by Geyer and Daniilidis in [9] and has been intensively used by the vision and robotics community (structure from motion, calibration, visual servoing, ...). Let us outline the essential of this model. Consider a virtual unitary sphere centered in M as shown in Fig. 2 and the perspective camera centered in C. The frames attached to the sphere and the perspective camera are related by a simple translation of $-\xi$ along the Z axis. Let $\mathcal{X}$ be a 3D point with coordinates $\mathbf{X} = [X\ Y\ Z]^\top$ in $\mathcal{F}_m$. The world point $\mathcal{X}$ is projected in the image plane into the point of homogeneous coordinates $\mathbf{p} = \mathbf{Km}$, where $\mathbf{K}$ is a 3×3 upper triangular matrix containing the conventional camera intrinsic parameters coupled with mirror intrinsic parameters and

$$\mathbf{m} = [x \ y \ 1]^\top = \left[\frac{X}{Z + \xi\|\mathbf{X}\|} \ \frac{Y}{Z + \xi\|\mathbf{X}\|} \ 1 \right]^\top \tag{1}$$

The matrix $\mathbf{K}$ and the parameter ξ can be obtained after calibration using for examples the methods proposed in [10]. In the sequel, the central imaging system is considered calibrated. In this case, the inverse projection onto the unit sphere $\mathbf{X_m}$ can be obtained as

$$\mathbf{X_m} = \lambda \left[x \ \ y \ \ 1 - \frac{\xi}{\lambda} \right]^\top \tag{2}$$

where $\lambda = \dfrac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1}$.

2.2 Line and cylindrical leg observation

To control a serial and a parallel robot, projected lines extracted from central catadioptric images will be exploited as input to a visual servoing control loop. Let us first illustrate the projection model of lines in the image of a central imaging system using a leg of the Gough-Stewart Platform. A Gough-Stewart Platform has six cylindrical legs of varying length q_j $(j = 1 \ldots 6)$ attached to the base by spherical joints located at points $\mathbf{A}_j$, and to the moving platform by spherical joints located at points $\mathbf{B}_j$ (see Figure 1). The image of the j^{th} leg is defined by the projection onto the image plane of two lines ($\mathcal{L}_1^j$ and $\mathcal{L}_2^j$) as depicted on Figure 2. Let us note $\mathbf{n}_j^i = [n_{jx}^i \, n_{jx}^i \, n_{jx}^i]^\top$ $(i = 1, 2)$ the unitary vector orthogonal to the interpretation plane π_j^i defined by the line $\mathcal{L}_j^i$ and the principal projection center. The points $\mathbf{X}_m$ lying on the intersection between π_j^i and the sphere are then defined by:

$$\begin{cases} \|\mathbf{X}_m\| = 1 \\ \mathbf{n}_j^{i\top} \mathbf{X}_m = 0 \end{cases} \tag{3}$$

Using the spherical coordinates given by eq. (2), it can be shown that 3-D points lying on $\mathcal{L}_j^i$ are mapped onto points $\mathbf{m}$ lying on a conic curve Γ_j^i, which can be written:

$$\alpha_0 x^2 + \alpha_1 y^2 + 2\alpha_2 xy + 2\alpha_3 x + 2\alpha_4 y + \alpha_5 = 0 \tag{4}$$

with $\alpha_0 = n_{jx}^{i2} - \xi^2(1 - n_{jy}^{i2})$, $\alpha_1 = n_{jy}^{i2} - \xi^2(1 - n_{jx}^{i2})$, $\alpha_2 = n_{jx}^i n_{jy}^i (1 - \xi^2)$, $\alpha_3 = n_{jx}^i n_{jz}^i$, $\alpha_4 = n_{jy}^i n_{jz}^i$ and $\alpha_5 = n_{jz}^{i2}$. Let us note that (4) is defined up to a scale factor. If $\alpha_5 \neq 0$, the number of parameters, can be reduced and (4) can be written as:

$$\beta_0 x^2 + \beta_1 y^2 + 2\beta_2 xy + 2\beta_3 x + 2\beta_4 y + 1 = 0 \tag{5}$$

with $\beta_k = \frac{\alpha_k}{\alpha_5}$. From the parameters β_k, it is possible to determine the perpendicular vector to the interpretation plane as follows:

$$n^i_{jz} = (\beta_3^2 + \beta_4^2 + 1)^{-\frac{1}{2}} = b; \;\; n^i_{jx} = \beta_3 b; \;\; n^i_{jy} = \beta_4 b \tag{6}$$

The case where $\alpha_5 = 0$ corresponds to a degenerate configuration where the optical axis lies on the interpretation plane. The orientation of the j^{th} leg, expressed in the camera frame, can straightforwardly be computed from the related normal vectors:

$$\mathbf{u}_j = \frac{\mathbf{n}^1_j \times \mathbf{n}^2_j}{\|\mathbf{n}^1_j \times \mathbf{n}^2_j\|} \tag{7}$$

In the sequel, we will see how to exploit line projections to design vision-based control scheme.

3 Control

In few words, let us first recall that the time variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s} = [\mathbf{s_1}^\top, \mathbf{s_2}^\top, \cdots \mathbf{s_n}^\top]^\top$ (where $\mathbf{s_i}$ are m-dimensional vectors containing the visual observations at the current configurations of the robotic system) can be expressed linearly with respect to the relative camera-object kinematics screw $\boldsymbol{\tau}$ (containing the instantaneous angular velocity $\boldsymbol{\omega}$ and the instantaneous linear velocity $\mathbf{v}$ of the origin of $\mathcal{F}_m$ expressed in the mirror frame) by:

$$\dot{\mathbf{s}} = \mathbf{L}\boldsymbol{\tau} \tag{8}$$

where $\mathbf{L}$ is the interaction matrix related to $\mathbf{s}$. In order to control the movements of a robot from visual features, one defines a task function to be regulated to $\mathbf{0}$ as [11]:

$$\mathbf{e} = \widehat{\mathbf{L}}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{9}$$

where $\hat{\mathbf{L}}^{+}$ is the pseudo-inverse of a chosen model of the $(n.m) \times 6$ interaction matrix $\mathbf{L}$ and $\mathbf{s}^*$ the desired value of $\mathbf{s}$. A very simple control law can then be designed by trying to ensure a decoupled exponential decay of the task function [12, 5]:

$$\boldsymbol{\tau} = -\lambda \mathbf{e} = -\lambda \widehat{\mathbf{L}}^{+}(\mathbf{s} - \mathbf{s}^*). \tag{10}$$

In order to compute the control law (10), it is necessary to provide an approximated interaction matrix $\widehat{\mathbf{L}}$. In the sequel, we will first derive its analitical form when the camera (in eye-in-hand or hand-in-eye configuration) observe a target composed of a set of 3D lines and then we will consider the case of a camera observing the legs of a parralel robot.

3.1 Visual servoing of a serial robot

Let us first assume that the camera observe a set of lines $\mathcal{L}_k$ with Plücker coordinates $\begin{bmatrix}\mathbf{n}_k, \mathbf{u}_k\end{bmatrix}^\top$ in the mirror frame and define the observation vector $\mathbf{s}_k$ for a projected line (conic) in the central catadioptric image as:

$$\mathbf{s}_k = \begin{bmatrix}\beta_{k3} \; \beta_{k4}\end{bmatrix}^\top \tag{11}$$

and the observation vector for n conics as $\mathbf{s} = [\mathbf{s}_1^\top \cdots \mathbf{s}_n^\top]^\top$. For convenience, in this section, we consider only one line and the subscript k will be omitted. Since parameters β_i only depend on $\mathbf{n}$, we can write equation (8) as:

$$\dot{\mathbf{s}} = \mathbf{J_{sn}} \mathbf{L_n} \boldsymbol{\tau} \tag{12}$$

where:

- $\mathbf{L_n}$ is the interaction matrix related to the normal vector $\mathbf{n} = [n_x, n_y, n_z]^\top$ to the interpretation plane for line $\mathcal{L}_i$ expressed in the mirror frame (such that $\dot{\mathbf{n}} = \mathbf{L_n}\boldsymbol{\tau}$), and
- $\mathbf{J_{sn}} = \frac{\partial \mathbf{s}}{\partial \mathbf{n}}$.

The interaction matrix related to the observation vector $\mathbf{s}$ is $\mathbf{L} = \mathbf{J_{sn}}\mathbf{L_n}$. It can be shown that ([13, 14]):

$$\dot{\mathbf{n}} = \mathbf{L_n}\boldsymbol{\tau} = \frac{\mathbf{v}^\top \mathbf{n}}{h}(\mathbf{u} \times \mathbf{n}) - \boldsymbol{\omega} \times \mathbf{n}$$

where h is the orthogonal distance from $\mathcal{L}_k$ to the origin of the mirror frame. According to the previous equation, the interaction between the normal vector and the sensor motion is thus:

$$\begin{aligned} \mathbf{L_n} &= \left(\tfrac{1}{h}(\mathbf{u} \times \mathbf{n})\mathbf{n}^\top \ [\mathbf{n}]_\times \right) \\ &= \left(\tfrac{1}{h}[\mathbf{u}]_\times \mathbf{n}\mathbf{n}^\top \ [\mathbf{n}]_\times \right) \\ &= \left(\mathbf{U}_h \mathbf{N}_{.} \ \mathbf{N}_\times \right) \end{aligned} \tag{13}$$

where $\mathbf{N}_\times = [\mathbf{n}]_\times$ denotes the antisymmetric matrix associated to the vector $\mathbf{n}$, $\mathbf{N}_{.} = \mathbf{n}\mathbf{n}^T$, and $\mathbf{U}_h = \frac{1}{h}[\mathbf{u}]_\times$. Note that the matrices $\mathbf{N}_\times$ and $\mathbf{N}_{.}$ can be computed using the visual features $\mathbf{s}$ (refer to Equation (6)):

$$\begin{aligned} \mathbf{N}_\times &= b \begin{pmatrix} 0 & -1 & \beta_4 \\ 1 & 0 & -\beta_3 \\ -\beta_4 & \beta_3 & 0 \end{pmatrix} \\ \mathbf{N}_{.} &= b^2 \begin{pmatrix} \beta_3^2 & \beta_3\beta_4 & \beta_3 \\ \beta_3\beta_4 & \beta_4^2 & \beta_4 \\ \beta_3 & \beta_4 & 1 \end{pmatrix} \end{aligned} . \tag{14}$$

The Jacobian $\mathbf{J_{sn}}$ is obtained by computing the partial derivative of (11) with respect to $\mathbf{n}$ and using (6):

$$\mathbf{J_{sn}} = \frac{1}{b} \begin{pmatrix} 1 & 0 & -\beta_3 \\ 0 & 1 & -\beta_4 \end{pmatrix} . \tag{15}$$

By combining the equations (13) and (15) and according to relation (12), the interaction matrix $\mathbf{L}$ is :

$$\mathbf{L} = \left(\tfrac{1}{h\,b}\mathbf{A} \ \mathbf{B} \right) \tag{16}$$

where

$$\mathbf{A} = \begin{pmatrix} u_y\beta_3 & u_y\beta_4 & u_y \\ -u_x\beta_3 & -u_x\beta_4 & -u_x \end{pmatrix} \tag{17}$$

and

$$\mathbf{B} = \begin{pmatrix} \beta_3\beta_4 & -1-\beta_3^2 & \beta_4 \\ 1+\beta_4^2 & -\beta_3\beta_4 & -\beta_3 \end{pmatrix} \tag{18}$$

3.2 Visual servoing of a parallel robot: the Gough-Steward platform

Let us now consider that the omnidirectional camera observe the legs of the Gough-Stewart platform. To servo the platform, two kinds of visual features (leg directions and leg edges) will be exploited.

Visual servoing of leg directions. To servo the leg directions, we define $\mathbf{s}$ as the geodesic error between the current leg orientation $\mathbf{u}_j$ and the desired one $\mathbf{u}_j^*$:

$$\mathbf{s}_{\mathbf{u}j} = \mathbf{u}_j \times \mathbf{u}_j^*,\ j = 1..6 \tag{19}$$

This means that: $\mathbf{s}_{\mathbf{u}j}^* = \mathbf{0_{3\times 1}}, j = 1..6$. Following [7] the interaction matrix associated with a leg orientation $\mathbf{u}_j$ is given by:

$$\dot{\mathbf{u}}_j = \mathbf{M}_j\,\boldsymbol{\tau} \tag{20}$$

$$\mathbf{M}_j = -\frac{1}{q_j}\left[\mathbf{I}_3 - \mathbf{u}_j\mathbf{u}_j^\top\right]\left[\mathbf{I}_3\ -[\mathbf{A}_j + q_j\mathbf{u}_j]_\times\right] \tag{21}$$

By combining (20) and (19), the time derivative of $\mathbf{s}_{\mathbf{u}j}$ can be written:

$$\dot{\mathbf{s}}_{\mathbf{u_j}} = \mathbf{L}_{\mathbf{u_j}}\,\boldsymbol{\tau} \tag{22}$$

$$\mathbf{L}_{\mathbf{u_j}} = -[\mathbf{u}_j^*]_\times\mathbf{M}_j \tag{23}$$

Now, the standard method applies: we stack each individual errors $\mathbf{s}_{u_j}$ in a single over-constrained vector $\mathbf{s}_\mathbf{u}$ as well as each associated individual interaction matrix $\mathbf{L}_{\mathbf{u_j}}$ into a compound one $\mathbf{L}_u$ and impose a first-order convergence to $\mathbf{s}_\mathbf{u}$. This yields the following pseudo-control vector $\boldsymbol{\tau}$

$$\boldsymbol{\tau} = -\lambda\widehat{\mathbf{L}}_u^{+}\mathbf{s}_\mathbf{u} \tag{24}$$

Visual servoing of the interpretation planes. Another possible set of visual features to control the Gough-Steward platform is composed of the two edges of each cylinder leg. Contrary to the perspective case where the leg edge projection is a line (and can be represented by a simple change of coordinates of the interpretation plane), the general case requires to reconstruct the interpretation planes in the frame related to the sphere (i.e. the sphere defined in the camera unified model) from the image data, knowing the intrinsic parameters. More details about the interpretation planes reconstruction in the general case is given in [8]. Formally the features related to the interpretation planes are defined by:

$$\mathbf{s}_{\mathbf{n_j^i}} = \mathbf{n}_j^i \times \mathbf{n}_j^{i*},\ j = 1\ldots 6,\ i = 1, 2 \tag{25}$$

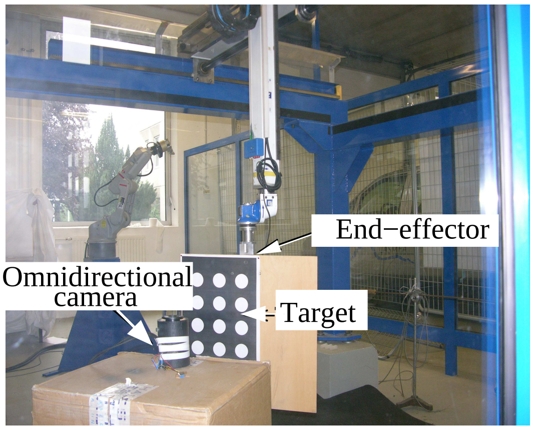


Fig. 3. Experimental setup : eye-to-hand configuration

The derivative of a leg edge, expressed in the camera frame can be obtained as described in [6]:

$$\dot{\mathbf{n}}_j^i = {}^{\mathbf{n}}\mathbf{J}_{\mathbf{u}}\mathbf{M}_i\boldsymbol{\tau} \tag{26}$$

$${}^{\mathbf{n}}\mathbf{J}_{\mathbf{u}} = \left[\frac{(\mathbf{u}_j \times \mathbf{n}_j^i)\mathbf{A}_j^\top}{\mathbf{A}_j(\mathbf{u}_j \times \mathbf{n}_j^i)^\top} - \mathbf{I}\right]\mathbf{u}_j\mathbf{n}_j^{i\top} \tag{27}$$

Consequently, by combining (27) and (25), the time derivative of $\mathbf{s}_{\mathbf{n}_j^i}$ can be written:

$$\dot{\mathbf{s}}_{\mathbf{n_j^i}} = \mathbf{L}_{\mathbf{n_j^i}}\,\boldsymbol{\tau} \tag{28}$$

$$\mathbf{L}_{\mathbf{n_j^i}} = -[\mathbf{n}_j^{i*}]_\times\,{}^{\mathbf{n}}\mathbf{J}_{\mathbf{u}}\mathbf{M}_i \tag{29}$$

This yields the following pseudo-control vector $\boldsymbol{\tau}$

$$\boldsymbol{\tau} = -\lambda\widehat{\mathbf{L}_n}^{+}\mathbf{s}_{\mathbf{n}} \tag{30}$$

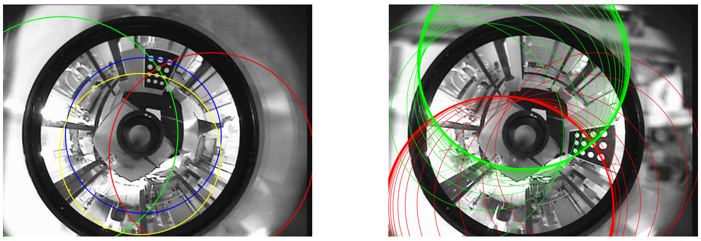
where $\mathbf{s}_{\mathbf{n}} = [\mathbf{s}_{\mathbf{n_1^1}}{}^\top\ \mathbf{s}_{\mathbf{n_1^2}}{}^\top \ldots \mathbf{s}_{\mathbf{n_6^2}}{}^\top]^\top$.

4 Experimental results

In this section, we present experimental results of central catadioptric visual servoing from lines for a 6 d.o.f robot manipulator and the parallel Gough-Stewart platform.

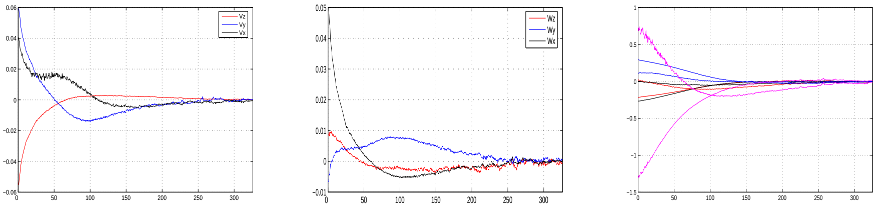
4.1 Visual servoing of a 6 dof serial robot

The proposed control law has been validated on a six d.o.f eye-to-hand system (refer to Figure 3). In this configuration, the interaction matrix has to take into



(a) (b)

Fig. 4. 2D Visual servoing from lines: (a) initial image and (b) desired image and trajectories of conics (for readability's sake, only trajectories of two conics are drawn).



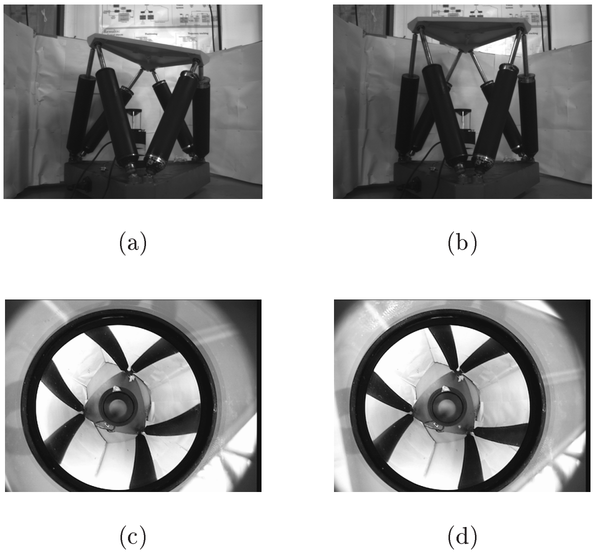
(a) (b) (c)

Fig. 5. Velocity and error vectors: (a) translational velocities [m/s], (b) rotational velocities [rad/s] and (c) image error $\mathbf{s} - \mathbf{s}^*$ versus iteration number.

account the mapping from the camera frame onto the robot control frame [15]. If we denote this mapping by $[\mathbf{R}_e,\ \mathbf{t}_e]$, the eye-to-hand interaction matrix $\mathbf{L}_e$ is related to the eye-in-hand one $\mathbf{L}$ by :

$$\mathbf{L}_e = \mathbf{L} \begin{bmatrix} \mathbf{R}_e & [\mathbf{t}_e]_\times \mathbf{R}_e \\ \mathbf{0}_3 & \mathbf{R}_e \end{bmatrix} \tag{31}$$

where $[\mathbf{t}_e]_\times$ is the skew symmetric matrix associated with translation vector $\mathbf{t}_e$. The interaction matrix $\mathbf{L}_e$ is used in the control law (10). Since we were not interested in image processing in this paper, the target is composed of white marks (see Figure 3) from which straight lines can be defined (see Figure 4(a)). The coordinates of these points (the center of gravity of each mark) are extracted and tracked using the VISP library [16]. The omnidirectional camera used is a parabolic mirror combined with an orthographic lens ($\xi = 1$). The image corresponding to the desired and initial configurations are given in Figures 4(a) and 4(b) respectively. The corresponding object displacement is composed of a translation $\mathbf{t} = [-10\ \ -80\ 60]^\top$ cm and a rotation (expressed as a rotational vector) $\theta\mathbf{u} = [0\ 0\ 100]^\top$ dg. The error between the visual features (desired and current) are plotted on Fig. 5(c) while the camera velocities are plotted on Fig. 5(a)-(b). These results confirm that the positioning task is correctly achieved.



(a) (b)

(c) (d)

Fig. 6. Experimental results: (a) initial configuration, (b) desired configuration, (c) initial image, (d) desired image

The trajectory of the conics in the image are plotted on 4(b) (for readability's sake, only trajectories of two conics are drawn).

4.2 Visual servoing of the Gough-Steward platform

In the following experiments, we give an example of an omnidirectional visual-servo of the Gough-Stewart platform (commercial DeltaLab *Table de Stewart* shown in Figure 6). The experimental robot has an analog joint position controller interfaced with Linux-RTAI. Joint velocity control is emulated through this position controller with an approximate 20ms sampling period. The omnidirectional camera used is a parabolic mirror combined with an orthographic lens. It is approximately placed at the base center. The projection of the legs in the image are almost radial. This property is used to detect the legs in the image. A set of circles centered on the principal point with diameters ranging from a minimal value d_{min} to a maximal value d_{max} are first defined. Next, the image is scanned along each circle providing a mono-dimensional signal which is then thresholded to obtain a binary signal. The peaks of the signal derivative (obtained using a gradient filter) provide then the image of the leg's edges. In theory, two circles are enough to determine each leg's edges in the image. In practice, more than two image points of each edge are required to obtain a robust estimation. For our experiments a set of 17 circles (which is a good compromise between robustness and time) with $d_{min} = 184\,pixel$ and $d_{max} = 370\,pixel$ are defined. Finally, note that the proposed method is fully automatic (no initialization by the user is required) and that less than $0.3ms$ are necessary to detect the leg edges with a conventional labtop.

For our experiments, the initial and desired configurations of the platform have been taken as shown on Figures 6.(a) and 6.(b). The corresponding images are given respectively on 6.(c) and 6.(d). In a first experiment, the leg directions were used to control the end-effector pose. Figure 7.(a) gives the behaviors of the feature error squares $\mathbf{s_i}^\top \mathbf{s_i}$. From this figure, we note that these errors decrease to 0. In a second experiment and for the same initial and desired robot configurations, the leg edges were used to control the end-effector pose. The same scalar gain λ was used for the first and second experiments. Figure 7.(b) shows that the system converges. However, plot of the feature errors are clearly smoother and less noisy than in Figure 7.(a). Furthermore, Figure 7(c) gives the plot of the variations of the leg orientation using leg orientation or leg edges as features in the control law. From this figure, it can be noticed that the variation of the orientation using leg edges (dashed plot) in the control is smoother and less noisy than using leg orientations (continuous plot).

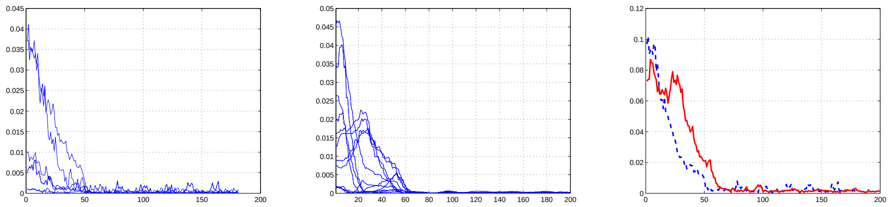


Fig. 7. (a) Experimental results using leg orientations ($\mathbf{s_{u_j}}$): errors $\mathbf{s_{u_j}}\mathbf{s_{u_j}}^\top$ (unitless) with respect to time (expressed as iteration number),(b) Errors $\mathbf{s_{n_j}}\mathbf{s_{n_j}}^\top$ (unitless) using the leg edges ($\mathbf{s_{n_j}}$) as visual features with respect to time, (c) Evolution of leg orientations during the control (sum of norms of the errors $\sum_{j=1}^{j=6} \|\mathbf{s_{u_j}}\mathbf{s_{u_j}}^\top\|$) with respect to time: results using leg orientation (control law (24), dashed plot) , results using leg edges (control law (30), continuous plot)

5 Conclusion

We have proposed two robotic applications of omnidirectional vision. More precicely, we have addressed the problem of controlling a robotic system (parallel and serial) by incorporating observations from a central catadioptric camera. We have validated the approach with a 6 d.o.f holonomic robot and a parallel Gough-Steward platform. The proposed approaches can be used with all central cameras (including conventional ones). More generally, the results presented in this paper extend the results obtained in the classical perspective case to the general case of the camera unified model. In future work, the analytical robustness and stability analysis with respect to the 3D parameters and calibration errors will be studied.

References

1. Blaer, P., Allen, P.: Topological mobile robot localization using fast vision techniques. In: IEEE International Conference on Robotics and Automation, Washington, USA (May 2002) 1031–1036
2. Winter, N., Gaspar, J., Lacey, G., Santos-Victor, J.: Omnidirectional vision for robot navigation. In: Proc. IEEE Workshop on Omnidirectional Vision, OMNIVIS, South Carolina, USA (June 2000) 21–28
3. Benosman, R., Kang, S.: Panoramic Vision. Springer Verlag ISBN 0-387-95111-3 (2000)
4. Baker, S., Nayar, S.K.: A theory of single-viewpoint catadioptric image formation. International Journal of Computer Vision **35**(2) (November 1999) 1–22
5. Hutchinson, S., Hager, G., Corke, P.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation **12**(5) (October 1996) 651–670
6. Andreff, N., Dallej, T., Martinet, P.: Image-based visual servoing of a Gough-Stewart parallel manipulator using leg observations. International Journal of Robotics Research **26**(7) (2007) 677–687
7. Andreff, N., Martinet, P.: Unifying kinematic modeling, identification and control of a Gough-Stewart parallel robot into a vision-based framework. IEEE Transactions on Robotics **22**(6) (2006) 1077–1086
8. Tahri, O., Mezouar, Y., Andreff, N., Martinet, P.: Omnidirectional visual-servo of a Gough-Stewart platform. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, California, USA (29 October to 2 November 2007)
9. Geyer, C., Daniilidis, K.: Mirrors in motion: Epipolar geometry and motion estimation. In: International Conference on Computer Vision, ICCV03, Nice, France (2003) 766–773
10. Mei, C., Rives, P.: Single view point omnidirectional camera calibration from planar grids. In: IEEE International Conference on Robotics and Automation. (April 2007)
11. Samson, C., Espiau, B., Borgne, M.L.: Robot Control : The Task Function Approach. Oxford University Press (1991)
12. Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation **8**(3) (June 1992) 313–326
13. Rives, P., Espiau, B.: Closed-loop recursive estimation of 3d features for a mobile vision system. In: IEEE Int. Conf. on Robotics and Automation, ICRA'87, Raleigh, Noth Carolina,USA (1987)
14. Andreff, N., Espiau, B., Horaud, R.: Visual servoing from lines. International Journal of Robotics Research **21**(8) (August 2002) 679–700
15. Flandin, G., Chaumette, F., Marchand, E.: Eye-in-hand / eye-to-hand cooperation for visual servoing. In: IEEE Int. Conf. on Robotics and Automation, ICRA'00. Volume 3., San Francisco, USA (April 2000) 2741–2746
16. Marchand, E., Spindler, F., Chaumette, F.: Visp for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robotics and Automation Magazine **12**(4) (December 2005) 40–52